

# LANL\*: Radiation Belt Drift Shell Modeling for Real-Time and Reanalysis Applications

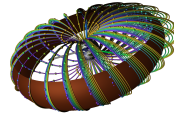


Martín Di Stefano, Josef Koller (mentor)  
ISR-1, Los Alamos National Lab  
email: [martind@lanl.gov](mailto:martind@lanl.gov), [jkoller@lanl.gov](mailto:jkoller@lanl.gov)

## Current problem motivating this work

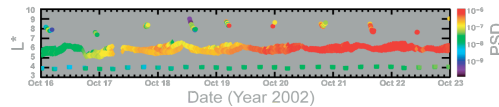
Physics modeling of radiation belts need adiabatic invariants ( $L^*$ ,  $\mu$ ,  $K$ ). In particular  $L^*$  is computationally expensive for more sophisticated magnetic field models.

$$L^* = -\frac{2\pi k_0}{\Phi R_E}$$



## Static dipole field not accurate enough

Modern research methods use data assimilation combining in-situ observations with models.  $L^*$  location of data changes with geomagnetic activity.



## More accurate models are too slow

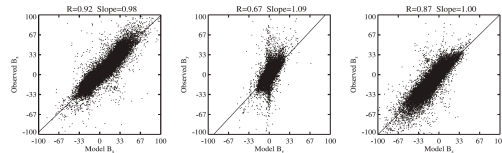
McCollough et al. (2008) compared computing times with different models. Compiler optimizations and parallelizing are possible but limited.

Model	Inputs	$L^*$ time (1440 values)
Tsyganenko et al. [2003]	Dst, p, By, Bz, G2, G3	2h 31m 07s
Tsyganenko [2002a, b]	Dst, p, By, Bz, G1, G2	3h 55m 35s
Tsyganenko [1995, 1996]	Dst, p, By, Bz	1h 43m 56s
Tsyganenko [1989]	Kp	4m 10s

## Accuracy of current field models are limited

Most popular models are empirical. Huang et al. (2008) report  $\Delta L^* \sim 13\%$  (quiet) and  $\Delta L^* \sim 50\%$  during storm times.

Inaccuracies of magnetic field models could alter radial phase-space density profiles of radiation belt electrons.



## Solution: a surrogate model using a neural network

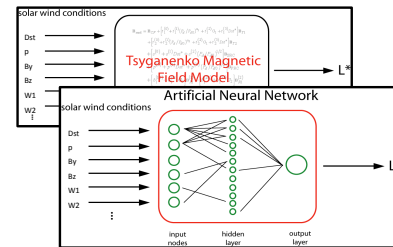
Surrogate models: polynomial regression, Kriging, neural network

- \* do not contain details about physical processes
- \* focus only on input-output relationship
- \* are by definition fast to compute

The results are not exact but sufficiently close to the original model used for training.

## LANL\* library replaces $L^*$ from Tsyganenko T04 with a much faster neural network

The results are sufficiently accurate while reducing the calculation time by 5-6 orders of magnitude.

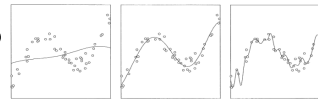


## The accuracy of neural network

is a function of number of nodes ( $N$ ) and training samples ( $M$ ). Cybenko 1989 proved that a sufficiently large neural network is able to approximate any function with arbitrary accuracy.

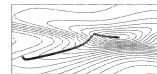
$$\mathcal{E} \propto O(1/\sqrt{N})$$

$$\mathcal{E} \propto O(1/M)$$



## Training of neural networks

automatically adjusts weights of interconnections to produce the desired outputs.



$$E = \frac{1}{PN} \sum_p \sum_i (t_{pi} - y_{pi})^2$$

Training a neural network is straightforward. Below is the matrix vector equation to be solved when training a single-hidden-layer neural network.

$$y = f^1 W^1 f^0 W^0 x + b^0 W^0 b^1$$

## LANL\* V2.0: a FORTRAN library

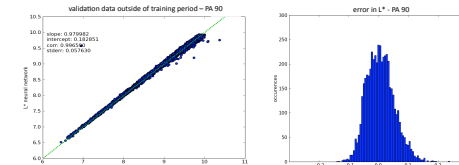
We have developed a prototype based on this technique. The prototype is based on the Tsyganenko (T04) magnetic field model.

The training data set consists of solar wind conditions over nearly an entire solar cycle – from 1995 through 2005.

10,000 spacecraft locations were randomly chosen over a 3D region surrounding the earth from 1.5 Re to 10 Re.

## Validation results show only small error by neural network technique

We validated the neural network with independent out-of-sample data and show that the error by the neural network is sufficiently small and typically less than 1%. Accuracy is reduced to a small degree with smaller pitch angles.

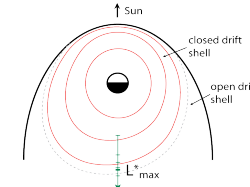


## LANL\* library is almost one million times faster

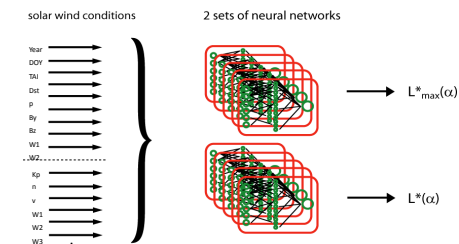
Our library provides a speedup of 5-6 orders of magnitude while providing sufficiently accurate  $L^*$  values.

## Future work: The last closed drift shell

is a function of pitch angle and requires a separate neural network.



The LANL\* library takes several solar wind conditions and the satellite coordinates to calculate  $L^*$  and the maximum valid  $L^*$  value. Additional parameters improved the trainability of the neural network.



## LANL\* V1.0 library is freely available

We have published the approach and the prototype LANL\* V1.0, which uses the Tsyganenko T01-storm model, in the Journal for Geoscientific Model Development. The library includes the complete neural network, Makefile for IDL and FORTRAN, and examples.

<http://www.geosci-model-dev.net/2/113/2009/gmd-2-113-2009.html>

A patent application describing this method is currently pending.

